StatesDataConfig:

1.- Data structure
Class that represents a particular geotemporal configuration. It stores which locations and times will have a desired correlation + monotonic pattern. It contains the following instance variables:

#REGARDING GEOGRAPHY
locations_total = 48
locations_noise_perc = 0
location_forced = None -> location that must has the desired pattern
location_range_forced = None -> location range (north, south, etc) that must have the desired pattern

#REGARDING TIME
times_total = 9
times_noise_perc = 0
time_forced = None
time_range_forced = None

It generates a dataframe with the following columns:
- State, name, x, y: state code, name and x,y values for its centroid.
- x_1-x_times_total, y_1-y_times_total: 2*times_total columns with the data values of the two variables that can be correlated: x and y.
- Pearson_r: referenced pearson of the initial data generated. It should be updated later but it is not done at the moment.
- N, S, W, E: columns that indicate with a one if a particular state belongs to a location_range.
- notnoise_geo: indicates if a particular state follows a particular pattern (=1) or not (=0)

It has one row per state + notnoise_time, that indicates if a particular time follow a particular pattern (=1) or not (=0).

A pattern will only be applied in which both columns of notnoise are equal to 1 at the same.
**Example:**

#REGARDING GEOGRAPHY
locations_total = 6
locations_noise_perc = 0.33
location_forced = None
location_range_forced = N

#REGARDING TIME
times_total = 6
times_noise_perc = 0.33
time_forced = None
time_range_forced = [2,4]

<table>
<thead>
<tr>
<th>State</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>y1</th>
<th>y2</th>
<th>y3</th>
<th>y4</th>
<th>y5</th>
<th>y6</th>
<th>North</th>
<th>notnoise_geo</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>notnoise_time</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In the above example:
- notnoise_time has ones in times 2, 3 and 4 (columns highlighted with yellow).
- not noise_geo has ones in states S1-S3, which match with the north. Rows highlighted with blue.

Therefore a pattern will be applied to the intersection of both, which is the zone in green.

2.- Add not noisy distractors to original configuration:

Rules:
- Always has 1/3 of noise, for both locations and time.
- When there is a range, location or time range, noise is applied outside the range.
- There are no contiguous rules. I.e, it is not necessary that a distractor is near a real not noise data.

In the following diagram lighter colors indicate distractors.

<table>
<thead>
<tr>
<th>State</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>y1</th>
<th>y2</th>
<th>y3</th>
<th>y4</th>
<th>y5</th>
<th>y6</th>
<th>North</th>
<th>notnoise_geo</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>notnoise_time</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CorrelatedDataGenerator:**

1.- Data generation algorithm

Given a StatesDataConfig generates correlated data. It has the following instance variables:

geotemp_config = config -> instance of StatesDataConfig
g = 0
monotonic = False
type_monotonic = None -> 1 indicates increasing, 0 indicates decreasing
r_exception = 0
monotonic_exception = False
type_monotonic_exception = None.
It follows the following algorithm:

**STEP 1.- Generate data to locations with notnoise_geo = 1.**
1.- Iterate through all states.
2.- For each state s, generate a pair $(x, y)$ of geotemp_config.times_total number of points with:
   2.A.- $r$, monotonic and type_monotonic in case the notnoise_geo column of that state equals 1
   2.B.- $r_{\text{exception}}$, monotonic_exception and type_monotonic_exception otherwise

<table>
<thead>
<tr>
<th>State</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>y1</th>
<th>y2</th>
<th>y3</th>
<th>y4</th>
<th>y5</th>
<th>y6</th>
<th>North</th>
<th>notnoise_geo</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**STEP 2.- Apply noise to temporal places with notnoise_time = 0:**
1.- Iterate through all states.
2.- For each state with notnoise_geo value = 1, generate a pair $(x, y)$ of
   geotemp_config.times_total number of points $r_{\text{exception}}$, monotonic_exception and
   type_monotonic_exception.
3.- Scale the new dataset with the minimum and maximum values of the initially computed
   dataset.
4.- switch values on those column in which notnoise_geo = 0

<table>
<thead>
<tr>
<th>State</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>y1</th>
<th>y2</th>
<th>y3</th>
<th>y4</th>
<th>y5</th>
<th>y6</th>
<th>North</th>
<th>notnoise_geo</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
2.- Mirror data generation algorithm

**Geographical shuffle:**

Case A: There is no location_range (zone) assigned.  
All locations are shuffled, ensuring that the location_forced follows the desired pattern (ie, notnoise_geo = 1).

Case B: There is location_range (zone) assigned.  
Locations inside the delimited location_range are shuffled among them. Locations outside a location_range are shuffled among them.

**Temporal shuffle:**

Case A: There is no time_range (period) assigned and no monotonic behavior  
All times are shuffled, ensuring that the time_forced follows the desired pattern (ie, notnoise_time = 1).

Case B: There is time_range (period) assigned and no monotonic behavior.  
Times inside the delimited time_range are shuffled among them. Times outside a time_range are shuffled among them.

Case C: There is monotonic behavior.  
Mirror it: if it is increasing, make it decreasing and vice versa.